2023/03/31 陈博喻

第一章 三. 1. 硬件系统 软件系统

3. 硬件资源 软件资源 5. 共享性 异步性

第一章 一、3. 临界区：访问共享资源的代码

4. 进程同步：异步环境下的一组并发进程因直接制约而互相发送消息、互相合作、互相等待，使得各进程按一定的速度执行的进程

二、1. A：就绪 B：执行 C：阻塞

3. A：用户登录 B：作业调度 C：提供服务 D：为进程分配CPU

三、1. 动态性 异发性 资源所有权 5. 动态 静态

6. 进程互斥 8. 共享内存 信道 消息队列

9. 执行 就绪 阻塞

5. S1没有前趋区 S2、S3的前趋区为S1 S4、S5的前趋区为S2 S6前趋区为S3
S7的前趋区为S4、S5、S6

```
semaphore a=0, b=0, c=0, d=0, e=0, f=0, g=0, h=0;
P1(){          P2(){          P3(){          P4(){
  ...            ...            ...            ...
  S1;            P(a);          P(b);          P(c);
  P(a);V(a);     S2;            S3;            S4;
  V(b);          V(c);          V(e);          V(f);
  ...            V(d);          ...            ...
}              }              }              }

P5(){          P6(){          P7(){
  ...            ...            ...
  P(d);          P(e);          P(f);
  S5;            S6;            P(g);
  V(g);          V(h);          P(h);
  ...            ...            S7;
}              }                ...
                              }

void main(){
  cobegin
  P1(); P2(); P3(); P4(); P5(); P6(); P7();
  coend
}
```

3. #include<bits/stdc++.h>

```
int in, out;        int o_count=0, a_count=0;
item fruit[5];
semaphore plate=1, full=0, empty=5, apple=0, orange=0;
void dad(){                          fruit=item
  while(true){fruit                  if(fruit==apple)
    produce an item nextp;             V(apple);
    P(empty);                          V(plate); else V(orange);
    P(plate);                        }
                                   }
```

```
void void son(){
    while(1){
        P(orangle);
        P(plate);
        V(empty);
        V(plate);
    }
}

void daughter(){
    while(1){
        P(apple);
        P(plate);
        V(plate);
        V(empty);
    }
}

void main(){
    cobegin
    dad(); son(); daughter();
    coend
}
```

4.

```
record x, x1, x2, x3;
semaphore mutex1=1, mutex2=1, empty1=0, empty2=0;
void PA(){ while(true){
    copy x from the dusk;
    P(empty1);
    P(mutex1);
    x1=x;
    V(mutex1);
}
}

void PB(){
    while(true){
        V(empty1); P(empty2);
        P(mutex1);
        P(mutex2);
        x2=x1; x1=0;
        V(mutex2);
        V(mutex1);
    }
}

void PC(){ while(true){
    P(mutex2);
    print x2;
    x2=0;
    V(empty2);
    V(mutex2);
}
}

void main(){
    cobegin
    PA(); PB(); PC();
    coend
}
```

5. semaphore customer=0, sever=0, mutex=0

```
int chair
void Customer(){
    while(1){
        P(mutex);
        if(chair>0){
            chair--;
            V(mutex);
            V(customer);
            P(sever);
            receive service;
        }
        else{
            V(mutex);
            leave;
        }
}
```

```
void Sever(){
    while(1){
        P(customer);
        P(mutex);
        chair--;
        V(mutex);
        V(sever);
        provide service;
    }
}
void main(){
    cobegin;
    Customer(); Sever();
    coend;
}
```

第三章 一、2.处理机调度：按照某种策略从就绪队列中选择一个进程，并分配CPU资源使其执行

3.周转时间：进程从提交到完成所需的总时间

4.死锁：多个进程（或线程）因竞争资源而陷入无限等待的状态

二、2.A:(2) B:(5) C:(1) D:(3) E:(4) F:(6)

4.A:(3)(4) B:(1)(2)(4)

三、2.提交、后备、运行、完成

6.预防 避免 检测 解除

1.解：FCFS：完成时间：A:3 B:9 C:13 D:18 E:20

周转时间：A:3 B:7 C:9 D:12 E:12

带权周转时间：A:1 B:$\frac{7}{6}$=1.17 C:2.25 D:2.4 E:6

平均周转时间：T=$\frac{1}{5}$×(3+7+9+12+12)=8.6

平均带权周转时间：T=$\frac{24}{7}$×$\frac{1}{5}$=$\frac{24}{35}$=7.8

非抢占式SJF：完成时间：A:3 B:9 C:15 D:20 E:11

周转时间：A:3 B:7 C:11 D:14 E:3

带权周转时间：A:1 B:$\frac{7}{6}$=1.17 C:3.75 D:4 E:5.5

平均周转时间：T=$\frac{1}{5}$×(3+7+11+14+3)=7.6

平均带权周转时间：T=$\frac{1}{5}$×9.25=1.84

抢占式SJF：完成时间：A:3 B:15 D:8 C:8 D:20 E:10

周转时间：A:3 B:13 C:4 D:14 E:2

带权周转时间：A:1 B:2.5 C:2 D:4 E:5

平均周转时间：T=$\frac{1}{5}$×(3+13+4+14+2)=7.2

平均带权周转时间：T=$\frac{1}{5}$×7.97=1.59

HRRN: 完成时间: A: 3 B: 9 C: 20 D: 16 E: 11

周转时间: A: 3 B: 7 C: 16 D: 10 E: 3

带权周转时间: A: 1 B: $\frac{9}{7}$=1.29 C: 4 D: 2 E: 1.5

平均周转时间: T=$\frac{1}{5}$×(3+7+16+10+3)=7.8

平均带权周转时间: T=$\frac{1}{5}$×(3+7+$\frac{9}{7}$+4+2+1.5)=1.96

RR: 完成时间: A: 13 B: 23 C: 19 D: 22 E: 17

周转时间: A: 13 B: 21 C: 15 D: 16 E: 9

带权周转时间: T=5, A: $\frac{13}{3}$=4.33 B: 3.5 C: 3.75 D: 3.2 E: 4.5

平均周转时间: T=$\frac{1}{5}$×(13+21+15+16+9)=14.8

平均带权周转时间: T=$\frac{1}{5}$×($\frac{13}{3}$+3.5+3.75+3.2+4.5)=3.86

2、T=0 ms A: 20-10-0=10

B: 50-10-0=40

C: 50-15-0=35 A的权为优先度最低，调度A执行

T=10 ms A: 40-10-10=20

B: 50-10-10=30

C: 50-15-10=25 A的权为优先度最低，调度A执行

T=20 ms A: 60-10-20=30

B: 50-10-20=20

C: 50-15-20=15 调度C执行

T=30 ms A: 60-10-30=20

35 35

B: 50-10-$\frac{30}{35}$=10.5

C: 5100-15-$\frac{30}{35}$=5550 调度B执行

3.(1) Allocation = $\begin{matrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ P_4 \end{matrix}\begin{pmatrix} 0 & 0 & 3 & 2 \\ 1 & 0 & 0 & 0 \\ 1 & 3 & 5 & 4 \\ 0 & 3 & 3 & 2 \\ 0 & 0 & 1 & 4 \end{pmatrix}$ 安全，按照$(P_0, P_3, P_1, P_4, P_2)$

(2) 能，在P_3执行完另外4个进程后分配

(3) 会